



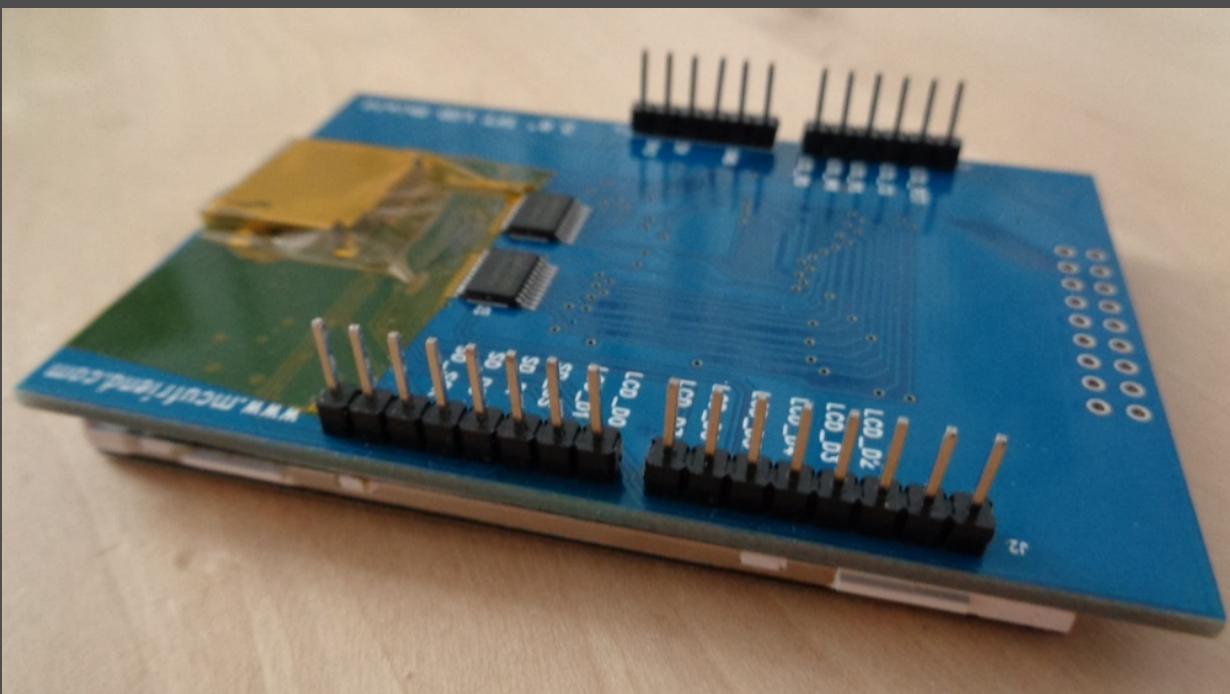
Arduino UNO TFT Display Test

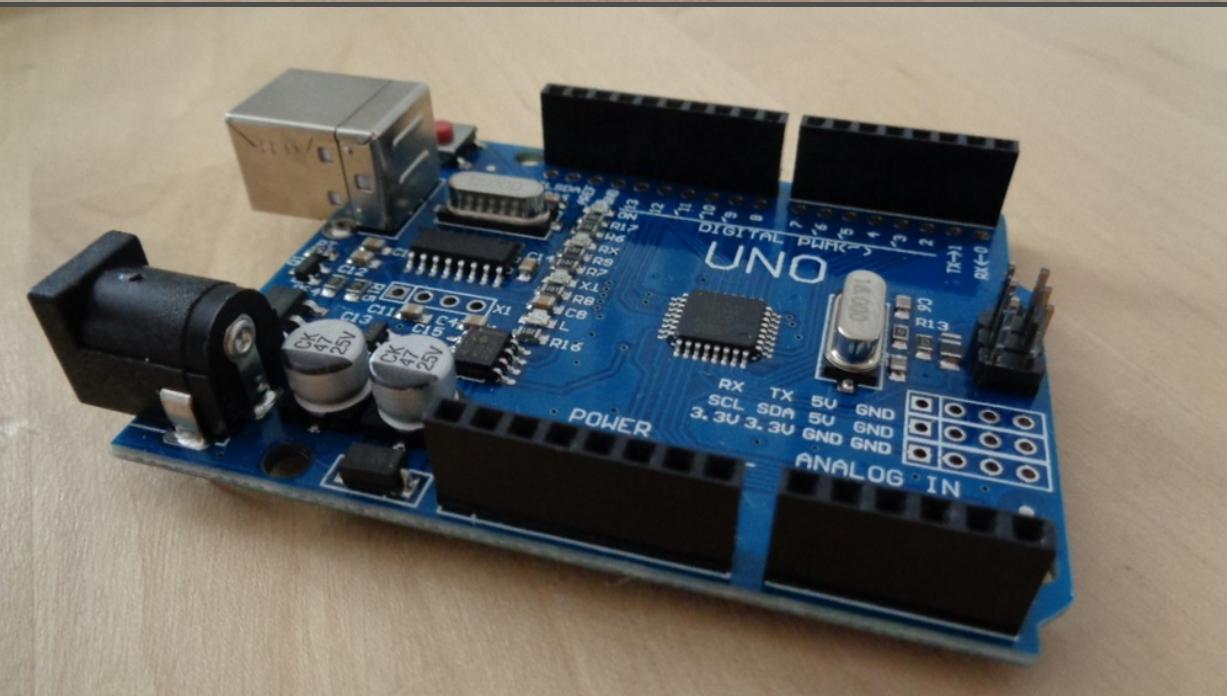
Preface

Since I am preparing the next version for the Cyberdeck, I have to do a display test. I want to replace the LCD of the [Cyberdeck version 5](#) with a TFT display. I have already prepared a [TFT shield helper](#) for it. But first I have to learn how everything works exactly and I prefer to do a dry run to be able to process errors faster. Therefore we learn here in this tutorial how to combine the TFT Display with our Arduino UNO and how to upload the source code. It is always better to do a few tries beforehand, because there is nothing worse than to despair of some errors during the actual assembly and then after much too much wasted time to have no more interest in the project.

Materials

Actually, we only need two main components. One is the mcufriend TFT Shield, which can be ordered on the Internet. In addition there is the Arduino UNO, which is available in almost every online shop. A suitable USB cable and a computer.



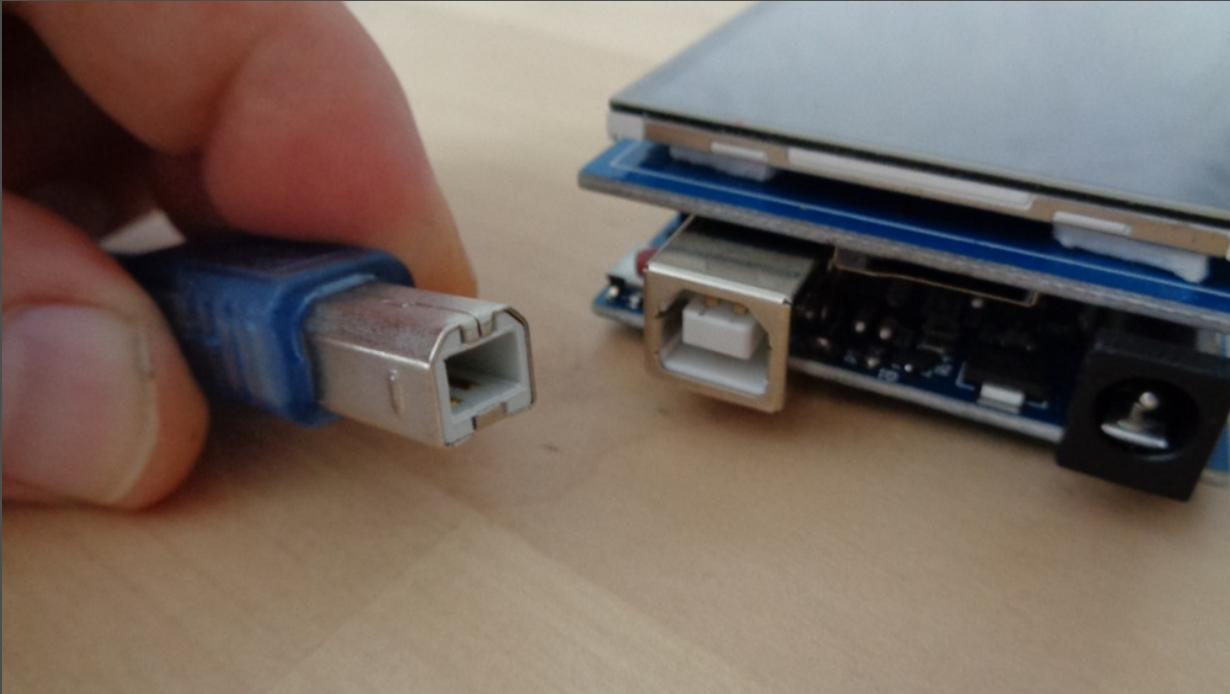


- Arduino UNO
- USB Cable

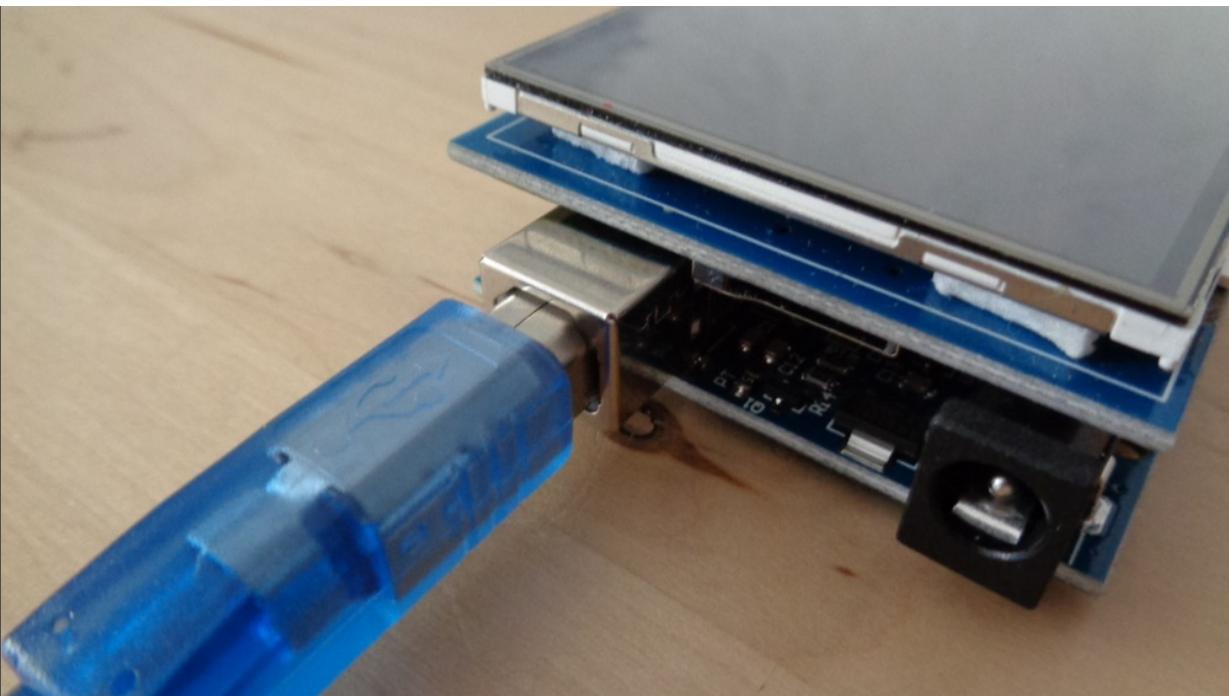
- mcufriend TFT Display 240x320

Assemble Hardware

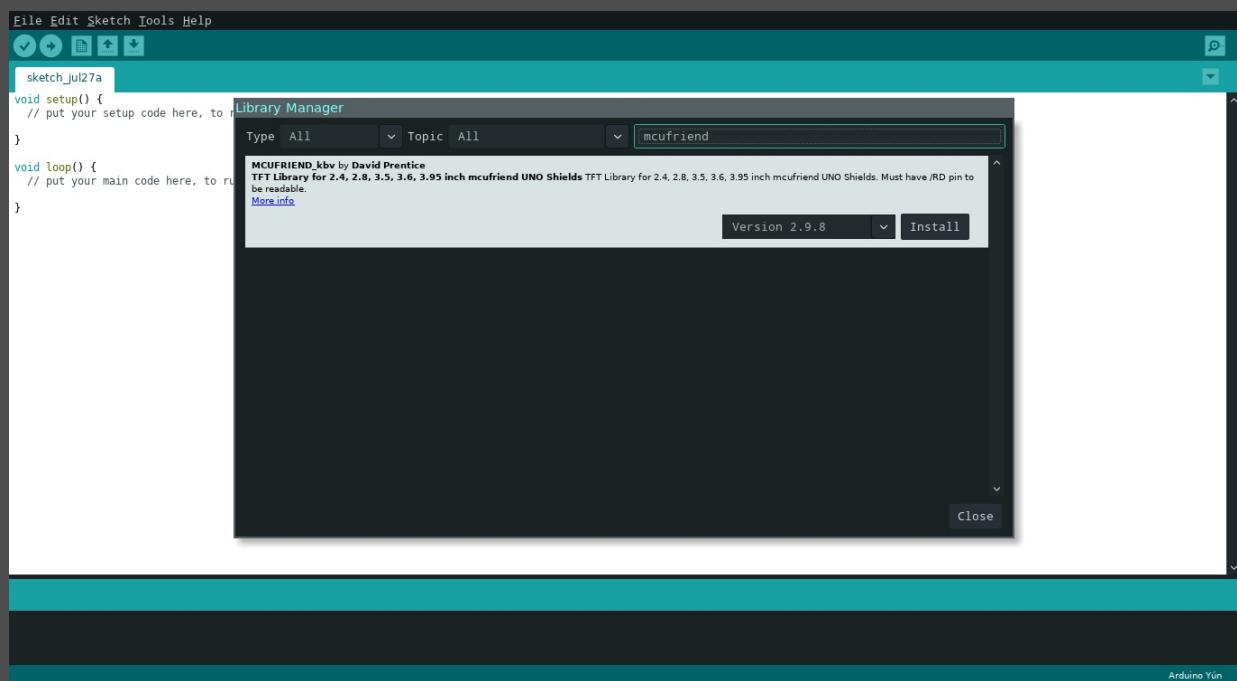
First we plug the TFT display onto the Arduino UNO. We make sure that the display looks like on the photo after the assembly. We also make sure that the tiller doesn't bend, which can happen very fast.



Then we plug the USB cable into the corresponding port and connect the other end to our computer.



Arduino UNO



We install the [Arduino IDE](#) **not** via the package manager but [download the current version](#) from the website. With the versions from the package managers it comes mostly to strange and hardly understandable error messages, which can be saved. We go to [Tools > Manage Libraries](#)" and search for `mcufriend`. We install the `MCURIEND_kbv` by [David Prentice](#). Then close and restart the Arduino IDE. How to set the ports and boards in the Arduino IDE is described in the [Cyberdeck v2 documentation](#).

The screenshot shows the Arduino IDE interface with the following details:

- File Edit Sketch Tools Help**: The top menu bar.
- Icons.c**: The current file being edited.
- graphictest_kbv**: The sketch name.
- Code Area:**

```
// All the mcufriend.com UNO shields have the same pinout.  
// i.e. control pins A0-A4, Data D2-D9, microSD D10-D13.  
// Touchscreens are normally A1, A2, D7, D6 but the order varies  
//  
// This demo should work with most Adafruit TFT libraries  
// If you are not using a shield, use a full Adafruit constructor()  
// e.g. Adafruit_TFTLCD tft(LCD_CS, LCD_CD, LCD_WR, LCD_RD, LCD_RESET);  
  
#define LCD_CS A3 // Chip Select goes to Analog 3  
#define LCD_CD A2 // Command/Data goes to Analog 2  
#define LCD_WR A1 // LCD Write goes to Analog 1  
#define LCD_RD A0 // LCD Read goes to Analog 0  
#define LCD_RESET A4 // Can alternately just connect to Arduino's reset pin  
  
#include <SPI.h> // f.k. for Arduino-1.5.2  
#include "Adafruit_GFX.h" // Hardware-specific library  
#include <MCUFRIEND_kbv.h>  
MCUFRIEND_kbv tft;  
//#include <Adafruit_TFTLCD.h>  
//Adafruit_TFTLCD tft(LCD_CS, LCD_CD, LCD_WR, LCD_RD, LCD_RESET);  
  
// Assign human-readable names to some common 16-bit color values:  
#define BLACK 0x0000  
#define BLUE 0x001F  
#define RED 0xFF00  
#define GREEN 0x07E0  
#define CYAN 0x07FF  
#define MAGENTA 0xF81F  
#define YELLOW 0xFFFF  
#define WHITE 0xFFFFF  
  
#ifndef min  
#define min(a, b) (((a) < (b)) ? (a) : (b))  
#endif  
  
void setup(void){  
}  
void loop(void){  
}
```
- Bottom Status Bar:** Arduino/Genuino Uno on /dev/ttyUSB0

Now we go to **File** > **Examples** > **MCUFRIEND_kbv** > **graphictest_kbv** to the required source code and load it into the IDE and compile it with **Sketch** > **Verify/Compile**. Then we can load the complicated code **Sketch** > **Upload** to the Arduino UNO.

Errors

It may happen that the error `Adafruit_GFX.h: No such file or directory` is added. We just go back to the library manager and install it. Sometimes you have to restart the Arduino IDE. Another bug that can occur on Linux is `can't open device /dev/ttyUSB0: Permission denied`. We can solve this by entering our terminal `sudo chmod +x a+rwx /dev/ttyUSB0`. After that everything should work normally again.